

CLAIMS

1. A method for upgrading managed state for a Java based application, comprising the operations of:

executing a Java module on a server, the Java module including at least one
5 original entity bean and at least one original state object in communication with the
original entity bean, the original state object storing a state of the original entity bean;

generating an upgraded state object;

transferring the state stored in the original state object to the upgraded state object;
and

10 providing state management for the original entity bean using the upgraded state
object.

2. A method as recited in claim 1, further comprising the operation of
creating an upgraded entity bean.

15

3. A method as recited in claim 2, further comprising the operation of
managing the state of the upgraded entity bean using the upgraded state object.

4. A method as recited in claim 3, wherein both the original entity bean and the original state object are disabled.

5. A method as recited in claim 1, wherein the upgraded state object is generated by upgrading a physical schema using data stored in a repository.

6. A method as recited in claim 5, wherein functionality of the Java module is not disrupted when the upgraded state object is generated.

7. A method as recited in claim 5, wherein functionality of the Java application is not disrupted when the Java module is upgraded.

8. A Java platform capable of performing an online upgrade on a Java application, the Java platform comprising:

a Java module including at least one original entity bean and at least one original state object in communication with the original entity bean, wherein the original state object storing a state of the original entity bean, and wherein the state object provides state management for the original entity bean; and

a repository having upgraded class files for the original entity bean and upgraded class files the original state object,

wherein the original state object is upgraded by generating an upgraded state object using upgraded class files from the repository, and transferring the state stored in the original state object to the upgraded state object.

5 9. A Java platform as recited in claim 8, wherein an upgrade entity bean is created when upgrading the Java platform.

10 10. A Java platform as recited in claim 9, wherein the state of the upgraded entity bean is managed using the upgraded state object.

15 11. A Java platform as recited in claim 10, wherein both the original entity bean and the original state object are disabled.

 12. A Java platform as recited in claim 8, wherein the upgraded state object is
15 generated by upgrading a physical schema using data stored in the repository.

 13. A Java platform as recited in claim 12, wherein functionality of the Java module is not disrupted when the upgraded state object is generated.

14. A Java platform as recited in claim 13, wherein functionality of the Java application is not disrupted when the Java module is upgraded.

15. A method for upgrading a Java application having a managed application
5 state, comprising the operations of:

executing a Java module on a server, the Java module including at least one original entity bean and at least one original state object in communication with the original entity bean, the original state object storing a state of the original entity bean;

generating an upgraded state object using data stored in a system repository;

10 transferring the state stored in the original state object to the upgraded state object;
and

providing state management for the original entity bean using the upgraded state object;

generating an upgraded entity bean using data stored in a system repository;

15 providing state management for the upgraded entity bean using the upgraded state object; and

disabling both the original entity bean and the original state object.

16. A method as recited in claim 15, wherein the upgraded state object is generated by upgrading a physical schema using data stored in a repository.

17. A method as recited in claim 16, wherein functionality of the Java module
5 is not disrupted when the upgraded state object is generated.

18. A method as recited in claim 17, wherein functionality of the Java application is not disrupted when the Java module is upgraded.

10 19. A method as recited in claim 18, wherein both the original state object and the upgraded state object are classified into a particular state management unit.

20. A method as recited in claim 19, wherein the particular state management unit is used to facilitate upgrading of the original state object.

15